

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-034619

(43)Date of publication of application : 09.02.2001

(51)Int.Cl.

G06F 17/30

(21)Application number : 11-203908

(71)Applicant : FUJITSU LTD

(22)Date of filing : 16.07.1999

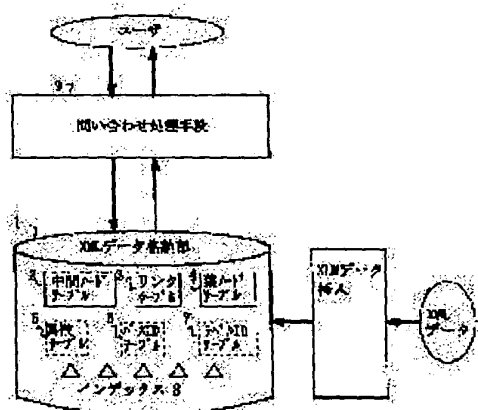
(72)Inventor : KANEMASA YASUHIKO
KUBOTA KAZUMI
ISHIKAWA HIROSHI

(54) STORE AND RETRIEVAL METHOD OF XML DATA, AND XML DATA RETRIEVAL SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To make storable XML data into a data base and to make executable a complicated inquiry at a high speed.

SOLUTION: A relation data base of an XML data store means 1 includes an intermediate node table 2 which stores the intermediate node information, a link table 3 which stores the link information, a leaf node table 4 which stores the leaf nodes, an attribute table 5 which stores the attribute information, a path ID table 6 where the path IDs are made to correspond to the character strings and a label ID table 7 where the label IDs are made to correspond to the character strings. The XML data which are expressed in a tree structure are divided into nodes, and these nodes are made to correspond to the link information and stored in the tables 2-7. When the XML data are retrieved, an inquiry statement is given to an inquiry processing means 9. The means 9 executes an inquiry to track a tree structure by using index 8 and outputs a requested retrieval result.



LEGAL STATUS

[Date of request for examination]

06.09.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

表し、四角い葉ノードはタグに付けられた属性(attribute)を表している。値を持つのはこの2つの葉ノードだけである。ノードを分割してデータベースに格納するとき、ノードの情報だけをデータベースのテーブルに格納したのでは、木構造のノード間の繋がりが、つまりリンクの情報が失われてしまう。そこで、リンクの情報はリンクの情報としてそれを格納する専用のテーブルを用意する。またノードも、中間ノードと、エレメント値の葉ノード、属性の葉ノードとは異なる格納構造が必要なので、別々のテーブルに格納する必要がある。

[0016] 本実施例で使用するテーブルは、全部次の6つである。

①中間ノードテーブル

これは中間ノードの情報を格納するテーブルである。ノードID(id)の他に、そのノードが含まれている文書の文書ID(docid)、そのノードまでのルートからのフルパスのID(pathid)をカラムとして持っている。

②リンクテーブル

これはノード間のリンクを格納するテーブルである。ノードID(id)、リンクのラベル(タグ名)のID(labelid)、子ノードのノードID(child)、その子ノードの全兄弟ノード中での出現順序(ord:total order)、その子ノードの同ラベルを持つ兄弟ノード中での出現順序(ord:partial order)をカラムとして持っている。上記のように、リンクテーブル中にラベル(タグ名)のID(labelid)を付加することによりタグ名を指定してリンクを辿る問い合わせを高速に実行することが可能となる。

[0017] ③葉ノードテーブル

これはエレメント値の葉ノードを格納するテーブルである。そのエレメント値の中間ノードのノードID(id)の他に、エレメントの値(value)と、そのエレメント中でのその値が出現した順序(ord)をカラムとして持っている。このように、値を持つための葉ノードテーブルを、前記中間ノードテーブルとは別に設けることにより、値を格納するスペースを節約することができる。

[0018] ④属性ノードテーブル

これはタグに付けられた属性(例えば図8におけるbook year="1995")におけるyear)を格納するテーブルである。そのタグが含まれるエレメントにある中間ノードのノードID(id)の他に、属性名のID(attributeid)、属性値(attribute value)をカラムとする。なお、属性テーブルに関係データベースの制約機能を用いて、(id, label)の組がユニークという制約をかけることによって、[同一のタグ内では同一の属性名は出現してはならない]というXMLの属性に関する構文規則をチェックすることができる。また、本実施例で想定している木構造表現では、XMLのタグが木構造のリンクに相当するので、XMLのタグに付けられる属性は本来ならばリンクに付くべきである。しかし、図4では、属性はリンク

に対してではなく、その下のノードに付いている。これ

により、XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

目のid(=5)は図4において、"5"と記されたノードを示し、そのorderは"0"、またその葉ノードテーブルにおいて、例えば第1行目のid(=3)は図4において、"3"と記されたノードを示し、そのlabelは"year"に対応、その属性値(attribute value)は"1995"である。また、パスIDテーブル、ラベルIDテーブルにはそれぞれ、上記各テーブル中のpathid、labelに対応したそれぞれの文字列、ラベルの文字列が格納され、例えば、pathid="1"に対応した文字列は"year"、label="bib book publisher name"であり、また、例えばlabelid="1"に対応した文字列は"bib"である。

[0023] (2) インデックスの構成

本実施例においては、本来連結されていたはずの木構造のノードが、前記したように1つ1つに分割されて関係データベースのテーブルに格納されている。このため、木構造を辿る問い合わせが行なわれた場合、問い合わせで辿る部分のリンクを連結し直すためにジョイン操作が行なわれる。このジョイン操作を高速に行なえるようにインデックスを効果的に張っておく必要がある。また、問い合わせが行なわれる場合、検索条件として指定されるのは、エレメントの属性、パス、出現順序などである。それらの検索を高速に行なう必要がある。そこで、ここにもインデックスを用意しておく必要がある。

[0024] 図7に、上記図5、図6に示したテーブルに張ったインデックスの一覧を示す。このインデックスはB+treeで張っており、キーが情報の属性の組からなるインデックスは、その組の先頭からの部分的な属性の組で検索に用いることもできる。なお中間ノードテーブルに張ってあるインデックスでキーが(pathid, id)のもの、あるいはpathid単独で鍵わないように思われるかもしれない。しかしキーをpathidだけにする、同じキー値を持つエントリが多量に発生して、B+treeインデックスが機能しなくなる。上記のようにキー値をpathidとノードのID(id)の組とすることにより、キーの重複を無くすることができ、B+treeの検索を高速に行うことができる。

[0025] (3) 問い合わせの実行

前記したように、格納されたXMLデータに対する問い合わせは、例えばXMLデータの問い合わせ言語で行なわれる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

われる。XMLデータのための検索言語の一つとして使

検索語XQLがある。XQLによる問い合わせ文を、例により簡単に説明する。

[0026]

SELECT result:(<book title>

FROM book: bib book

WHERE <book author: lastname="Darven";

この問い合わせの意味は「bib book author: lastnameがDarvenであるようなbib bookについて、bib book titleを検索結果として得たい」という意味である。

10

[0027] 上記に示すように、問い合わせ文は大きく、SELECT、FROM、WHEREの3つの部分に別れている。

SELECTの部分では検索結果として得たいエレメントのプロジェクションを指定する。FROMの部分では検索の対象となるエレメントを指定している。WHEREの部分では検索条件のセレクションを指定する。上記のような問い合わせは前記したように、問い合わせ処理エンジン13で処理される。問い合わせ処理エンジン13では、上記のための構文を生成する。そして、該構文を基に、最適な実行プランを生成する。この実行プランは、木構造検索用の関数セットで記述される。

[0028] 次に、上記XMLデータに対する問い合わせ処理が、どのように行なわれるかを説明する。ここでは、図8のサンプルXMLデータを、XMLデータ格納部11に格納し、前述した図5、図6に示したテーブルに挿入した場合を例として、上記のように「著者がDarvenである本のタイトルを求めよ」という問い合わせを行なった場合について説明する。この場合のテーブル検索は、上記木構造検索用の関数により実行される。

[0029] 1. 葉ノードテーブルを検索して、値が"Darven"であるノードのノードID(=16)を得る。

2. パスIDテーブルを検索して、パス"bib book author: lastname"のパスID(=4)を得る。

3. 中間ノードテーブルを上記1. で得られたノードID(=16)で検索して、得られたパスID(=4)が上記2. で得られたパスID(=4)と一致することを確認する。

4. ラベルIDテーブルを検索して、ラベル"lastname"のラベルID(=8)を得る。

5. リンクテーブルを検索して、上記1. で得られたノードID(=16)と上記4. で得られたラベルID(=8)から、親ノードのノードID(=15)を得る。

6. ラベルIDテーブルを検索して、ラベル"author"のラベルID(=7)を得る。

7. リンクテーブルを検索して、上記5. で得られたノードID(=15)と上記6. で得られたラベルID(=7)から、親ノードのノードID(=9)を得る。

50

11

8. ラベルIDテーブルを構築して、ラベル"ille"のラベルID (=6) を得る。
9. リンクテーブルを構築して上記7. で得られたノードID (=9) と上記8. で得られたラベルID (=6) から、子ノードのノードID (=12) を得る。
10. 葉ノードテーブルを構築して、上記9. で得られたノードID (=12) から、そのノードの値("Foundation for Object/Relational Database") を得る。以上のようにして得られた検索結果は、問い合わせ処理エンジン13を介して出力され、ユーザに提示される。

【0030】

【発明の効果】 以上説明したように、本発明においては、関係データベースに、中間ノードの情報を格納するための中間ノードテーブルと、リンクの情報を格納するためのリンクテーブルと、葉ノードの情報を格納するための葉ノードテーブル等のテーブルを設け、XMLの木の構造をノードとリンクに分解して、上記テーブルに各ノードとリンク情報を関係付けて格納し、上記テーブルを参照して木構造を通る問い合わせを実行し、XMLデータを検索するようにしたので、データ構造が一意に定まっていなくてもXMLデータに対する複雑な問い合わせを高速に実行することができる。また、XMLの木構造をそのまま格納手段に格納するので、DTD無しのXMLデータや半構造のXMLデータも格納することができる。さらにXMLの木構造を全てデータベース上に格納しているため、木構造の全ての情報を検索に利用することができる。

【図面の簡単な説明】

【図1】 本発明の基本構成図である。

【図2】 本発明の実施例のシステムの構成例を示す図である。

12

【図3】 本発明の実施例のシステムにおける格納処理フローを示す図である。
【図4】 XMLデータの木構造表現の一例を示す図である。
【図5】 本発明の実施例のテーブル構成の一例を示す図(1)である。
【図6】 本発明の実施例のテーブル構成の一例を示す図(2)である。
【図7】 本発明の実施例のインデックス一覧を示す図である。

【図8】 XMLデータの一例を示す図である。

【図9】 図8のXMLデータをテーブルに格納した様子を示す図である。

【符号の説明】

- 1 XMLデータ格納格納手段
- 2 中間ノードテーブル
- 3 リンクテーブル
- 4 葉ノードテーブル
- 5 属性テーブル
- 6 パスIDテーブル
- 7 ラベルIDテーブル
- 8 インデックス
- 9 問い合わせ処理手段
- 11 XMLデータ格納部
- 12 XMLデータ格納モジュール
- 12a XMLパーザ
- 12b ロード
- 13 問い合わせ処理エンジン

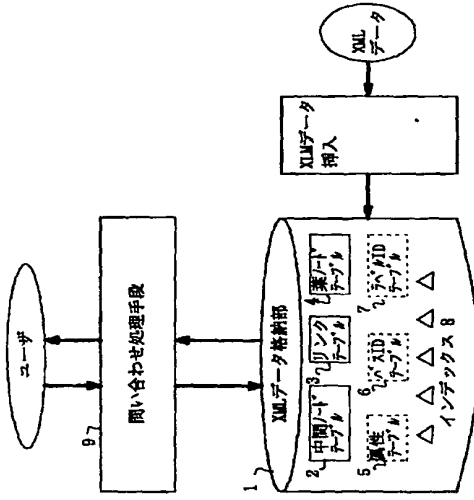
13a 問い合わせ処理のパーザ

13b 問い合わせ最適化エンジン

13c 木構造検索用

【図1】

本発明の基本構成図



【図7】

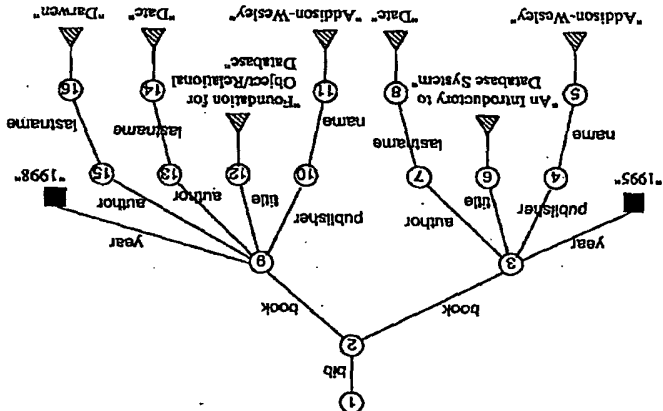
本発明の実施例のインデックス一覧を示す図

インデックス一覧

テーブル名	キー
中間ノード	id
中間ノード	(acid, id)
中間ノード	(getid, id)
リンク	(id, labelid, child)
リンク	(child, labelid, id)
葉ノード	id
葉ノード	value
属性ノード	id
属性ノード	attvalue
パスID	path
ラベルID	label

【図4】

XMLデータの本構造成現の一例を示す図



【図5】

本発明の実施例のテーブル構成の一例を示す図(1)

中間ノードテーブル			リンクテーブル			
id	decid	pathid	id	labelid	pard	child
5	1	1	4	5	0	5
4	2	2	7	8	0	8
6	1	3	3	4	0	4
8	1	4	3	6	1	6
7	1	5	3	7	2	7
3	1	6	10	5	0	11
11	1	1	13	8	0	14
10	1	2	15	8	0	16
12	1	3	9	4	0	10
14	1	4	9	6	1	12
13	1	5	9	7	2	13
16	1	4	9	7	3	15
15	1	5	2	2	0	3
9	1	6	2	2	1	9
2	1	7	1	1	0	2
1	1	8				

【図 6】

本発明の実施例のテーブル構成の一例を示す図 (2)
葉ノードテーブル

id	order	value
5	0	Addison-Wesley
6	0	An Introductory to Database Systems
8	0	Date
11	0	Addison-Wesley
12	0	Foundation for Object/Relational Database
14	0	Date
16	0	Darwen

属性ノードテーブル

id	labelid	attvalue
3	3	1995
9	3	1998

パスIDテーブル

pathid	path
1	bib book publisher name
2	bib book publisher
3	bib book title
4	bib book author lastname
5	bib book author
6	bib book
7	bib
8	/

ラベルIDテーブル

labelid	label
1	bib
2	book
3	year
4	publisher
5	name
6	title
7	a author
8	lastname

【図 8】

XML データの一例を示す図

[DTD]

```
<ELEMENT book (author?, title, publisher?)>
<!ATTLIST book year CDATA>
<ELEMENT article (author?, title, year?, (shortversion | longversion))>
<!ATTLIST article type CDATA>
<ELEMENT publisher (name, address?)>
<ELEMENT author (firstname?, lastname?)>
```

[XML データ]

```
<books>
  <book year="1995">
    <title> An Introductory to Database System </title>
    <author> <lastname> Date </lastname> </author>
    <publisher> <name> Addison-Wesley </name> </publisher>
  </book>
  <book year="1998">
    <title> Foundation for Object/Relational Database </title>
    <author> <lastname> Date </lastname> </author>
    <author> <lastname> Darwen </lastname> </author>
    <publisher> <name> Addison-Wesley </name> </publisher>
  </book>
</books>
```

【図 9】

図 8 の XML データをテーブルに格納した様子を示す図

book のテーブル

ID	title	author1 firstname	author1 lastname	author2 firstname	author2 lastname	publisher name	publisher address	year
1	An Introductory to Database System					Addison-Wesley		1995
2	Foundation for Object/Relational Database					Addison-Wesley		1998
?	?	?	?	?	?	?	?	?

author2 firstname	author2 lastname	publisher name	publisher address	year
	Darwen	Addison-Wesley		1995
?	?	?	?	?

フロントページの続き

(72)発明者 石川 博

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(15)

特開2001-34619

Fターム(参考) 5B075 ND36 PP23 QR00 QT06